

**Amendments to the Claims**

**This listing of claims will replace all prior versions, and listings, of the claims:**

1. (currently amended) A method for invoking multiple parallel instances of a same node comprising the steps of:

a) defining a multinode as a node that allows for activation of multiple parallel instances of a same work node ~~activation of a node~~;

b) at run time determining ~~a~~ the number of work nodes to be activated in the multinode based on an activation rule;

c) executing the number of work nodes in the multinode;

d) determining when the execution of the multinode is completed based on a termination rule; ~~and~~

e) when the execution of the multinode is complete, executing a successor node;  
and

f) when the execution of the multinode is not complete, processing continues at step c).

2. (currently amended) The method of claim 1

wherein determining the number of work nodes to be activated based on said ~~an~~ activation rule includes determining the number of works nodes to be activated based on said ~~an~~ activation rule based on the number of resources available.

3. (currently amended) The method of claim 1

wherein determining the number of work nodes to be activated based on said ~~an~~ activation rule includes determining the number of work nodes to be activated based on said ~~an~~ activation rule based on ~~the number of~~ elements in a vector.

4. (currently amended) The method of claim 1

wherein determining when the execution of the multinode is completed based on said ~~a~~ termination rule includes evaluating whether a multinode goal has been achieved,

and when the multinode goal has been achieved, terminating the execution of the multinode.

5. (currently amended) The method of claim 1

wherein determining when the execution of the multinode is completed based on ~~said-a~~ termination rule includes determining whether all work-service nodes in the multinode have been completed, and when all work-service nodes in the multinode have been completed, terminating the execution of the multinode.

6. (currently amended) The method of claim 1

wherein terminating the execution of the multinode includes canceling ~~the other service~~ nodes and proceeding to a successor node.

7. (currently amended) The method of claim 1 further comprising the step of

allowing flow to continue to ~~said-a~~ successor node when all activated work ~~invoked-service~~ nodes in the multinode have been completed.

8. (currently amended) The method of claim 1 wherein the step of

executing the multinode includes the step of providing each work node in the multinode with different input data for execution.

9. (currently amended) The method of claim 1 wherein the step of

executing the multinode includes the step of providing different attributes for each work node in the multinode.

10. (currently amended) The method of claim 9 wherein the attributes includes one of resource selection criteria, security, exception handling criteria, and deadlines for work node execution.

11. (currently amended) The method of claim 1 wherein the step of determining when the

execution of the multinode is completed based on said-a termination rule further includes the step of

specifying multinode termination by a condition;

checking the condition when one of the work nodes in the multinode terminates;

and

when the condition is satisfied, saida successor node is activated, and other work nodes in execution within the multinode are canceled.

12. (currently amended) A system for processing multinode definitions comprising:

a workflow engine for processing workflow definitions; and

a multinode handling facility coupled to the workflow engine for processing multinodes, determining ~~a~~-the number of multiple instances of same work nodes in one of the multinodes to be activated based on an activation rule; executing the number of work nodes in the multinode; determining when the execution of the multinode is completed based on a termination rule; and when the execution of the multinode is complete, executing a successor node.

13. (currently amended) The system of claim 12 wherein the multinode handling facility further comprises

a multinode determination unit for receiving a node definition and responsive thereto for determining whether the current node is a normal work node or one of the-a multinodes.

14. (currently amended) The system of claim 12 wherein the multinode handling facility further comprises

an activation facility for receiving said-an activation rule and based thereon for determining whether activation is by resource or by variable.

15. (currently amended) The system of claim 14 wherein the multinode handling facility further comprises

a resource-based activation facility coupled to the activation facility for processing activation by resources; and

a variable-based activation facility coupled to the activation facility for processing activation by variables.

16. (currently amended) The system of claim 15 wherein the resource-based activation facility further comprises

a resource rule execution unit for executing a~~the~~ resource rule of the multinode; and

a new instance generation unit for starting new instances of the multinode for each new resource in the resource list.

17. (original) The system of claim 16 wherein the resource rule is specified in a service node tag of a multinode description.

18. (original) The system of claim 15 wherein the variable-based activation facility further comprises

a variable name reader for reading the variable name; and

a new instance generation unit for starting new instances of the multinode for each new element in the variable identified by the variable name.

19. (currently amended) The system of claim 18 wherein a~~the~~ variable type is one of a vector and a list.

20. (original) The system of claim 12 wherein each node in the multinode is provided with different input data and different attributes for execution; wherein the attributes includes one of resource selection criteria, security, exception handling criteria, and deadlines for node execution.